

Improved Confidence Estimates for Black-Box Large Language Models

Sokhna Diarra Mbacke
Layer 6 AI

diarra@layer6.ai

Mouloud Belbahri
TD Insurance

mouloud.belbahri@td.com

Gabriel Loaiza-Ganem
Layer 6 AI

gabriel@layer6.ai

Abstract

Uncertainty quantification (UQ) is essential for the safe deployment of large language models (LLMs). Existing methods, from verbalized confidence to ones requiring multiple generations, are often zero-shot and produce scores quantifying uncertainty without the need for labelled data. Nonetheless, in practice one must always evaluate their performance on a dataset of interest before deployment. In this work we show that, by leveraging this dataset, we consistently outperform these existing scores. Specifically, we build simple classifiers that predict LLM response correctness by using these scores and the correctness of similar queries as features. Our method produces minimal computational overhead, making it a cheap and straightforward enhancement for UQ in LLMs for real-world applications.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has led to their integration across diverse sectors, from creative writing (Gómez-Rodríguez & Williams, 2023) to automated code generation (Wang et al., 2021; Chen et al., 2021), and more (Chaturvedi & Verma, 2023). However, their susceptibility to hallucinations (Maynez et al., 2020; Xu et al., 2024) remains a significant barrier to deployment in high-stakes environments such as healthcare (Begoli et al., 2019) and scientific research (Wang et al., 2023a). Uncertainty Quantification (UQ) aims to address this challenge by evaluating model outputs to determine the degree to which a specific generation can be relied upon. Ultimately, establishing robust UQ frameworks is essential for unlocking the full potential of LLMs in many critical applications.

The field of UQ for LLMs is still largely open (Shorinwa et al., 2025), and no clear consensus exists over what exactly should be quantified in the first place. For example, the traditional split of uncertainty into aleatoric and epistemic (Hüllermeier & Waegeman, 2021)—a staple of supervised learning—has been criticized as insufficient for the unique complexities of LLMs (Kirchhof et al., 2025). In practice, two major trends exist in the field. The first, which we refer to as *heuristic scoring*, produces abstract uncertainty scores designed solely to correlate with response correctness. These scores serve primarily as abstention tools, with the model withholding its response whenever the score fails to meet some predefined threshold. The second, *confidence estimation*, attempts to directly estimate the probability of response correctness.

We argue that confidence should be the score of interest for UQ in LLMs: it subsumes heuristic scoring by maintaining the ability to trigger abstentions, while being more useful for downstream decision making. For example, calibrated probabilities of correctness can be integrated into decision-theoretical frameworks (Russell & Norvig, 2010; Berger, 2013), enabling systems to achieve optimal behaviour through the maximization of expected utility.

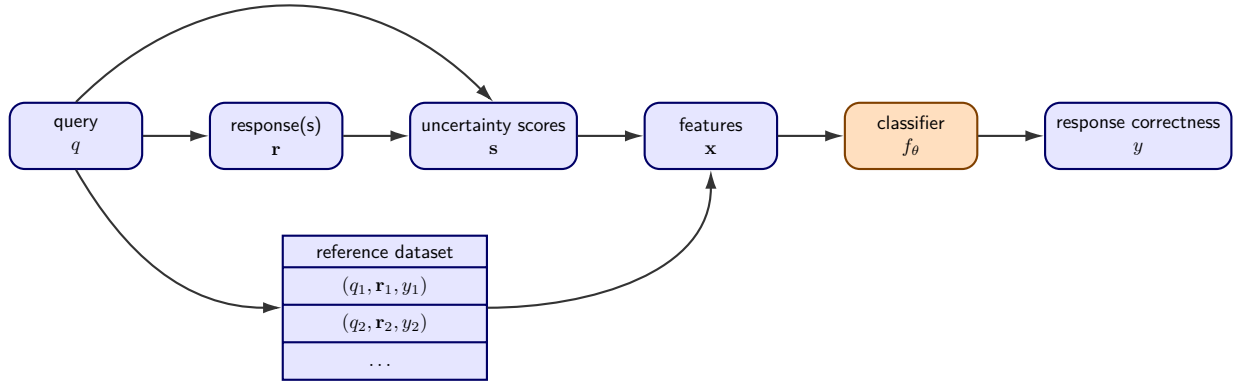


Figure 1: Most UQ methods use a given query q and one or more corresponding LLM-generated responses $\mathbf{r}$ to obtain uncertainty scores $\mathbf{s}$ whose goal is to correlate with y (a correctness label for the response). Instead, we retrieve similar queries from an existing dataset, and use the retrieved queries, alongside $\mathbf{s}$, to produce features $\mathbf{x}$; we then train a classifier predicting y from $\mathbf{x}$.

In order to provide better confidence estimates than are currently available, we first observe that existing UQ methods for LLMs operate predominantly in a zero-shot manner, generating uncertainty scores without requiring a training dataset. In practice, however, models are rarely deployed without being evaluated on a representative labelled dataset. Our work begins with the insight that this data is a significant, yet untapped resource; by moving beyond the zero-shot paradigm, we can leverage this data to improve the reliability of LLM uncertainty estimates.

More specifically, we partition this dataset into a training set and a *reference set*. To enhance any given uncertainty scores, we augment each query in the training data with these scores as features. We then leverage the reference set to engineer additional features: for every query in the training set, we identify its k -nearest neighbours within the reference set and extract neighbourhood-based statistics, such as the correctness of those neighbours and their cosine similarities to the query in embedding space. We then cast confidence estimation as a supervised binary classification task, leveraging the augmented training set to predict response correctness. We depict this process in Figure 1.

Our framework offers several distinct advantages. First, the computational overhead is minimal: training the classifier is efficient due to its low complexity, and the marginal cost of inference is negligible relative to the cost of generating an LLM response. Second, the method requires only a dataset which, as previously noted, is a standard prerequisite in practical scenarios, even for zero-shot methods. Finally, our approach is inherently flexible; it provides a mechanism to aggregate several uncertainty scores into a single one, and it can transform heuristic scores into confidence estimates.

Empirically, our classifiers consistently outperform existing UQ methods across various datasets and models, and provide calibrated confidence estimates. Our results also reveal that while our data-driven approach consistently improves upon baseline uncertainty scores, the optimal classifier model is task-dependent. This variability underscores our central thesis: while labelled data is indeed a vital, untapped resource for UQ, leveraging it effectively necessitates the selection of a task-specific learner to account for different data distributions.

2 Related Work and Motivation

Heuristic scoring. Although some heuristic scores are based on model logits (Duan et al., 2024), the majority rely on a sample-and-aggregate framework: multiple LLM responses are generated for a single query, and an uncertainty score is derived by summarizing the variability of these outputs into a scalar value. These methods diverge primarily in their mechanisms for generating responses and their specific summarization techniques. Wang et al. (2023b) use low-rank adaptation (LoRA; Hu et al., 2022) to fine-tune an ensemble of LLMs to produce varied responses, and Yang et al. (2024a) use instead a Laplace

approximation (Ritter et al., 2018) on the LoRA blocks. Hou et al. (2024) generate the responses by changing the query with different clarifications, and Gao et al. (2024) with different perturbations. Ross et al. (2025) treat the system prompt as a Bayesian parameter, and responses correspond to posterior samples. Other methods generate responses independently. Kuhn et al. (2023) group them into semantic clusters, and use the entropy of the corresponding distribution over clusters as the uncertainty score. Lin et al. (2024) instead produce a matrix containing pairwise similarities for the generated responses, and produce various uncertainty scores based on the eigenspectra and traces of related matrices; Nikitin et al. (2024) use the corresponding von Neumann entropy instead. Grewal et al. (2024) produce the score by averaging all the pairwise cosine similarities between generated responses, and Qiu & Miikkulainen (2024) by using the density obtained from a kernel density estimator fitted on the embedding space of the responses.

Confidence estimation. Verbalized confidence scores, often denoted as P_{true} , are obtained by directly prompting the LLM to assess the probability of its own response being correct (Kadavath et al., 2022; Yang et al., 2024b). However, these self-reported probabilities are prone to exhibit overconfidence (Zhou et al., 2024). While Kapoor et al. (2024) propose a fine-tuning strategy to calibrate these verbalized outputs, this procedure is computationally expensive and requires white-box access to the LLM. Other works have explored the use of classifiers to predict response correctness, though with significant differences from our approach. For instance, Mielke et al. (2022) train the classifier using the internal representations of the LLM as features, which requires white-box access to the LLM. The closest work to ours is APRICOT (Ulmer et al., 2024), where a classifier predicting response correctness is also trained. Ulmer et al. (2024) use an auxiliary model to embed queries and responses, and use the resulting embeddings as features. Their framework remains distinct from ours in how features are constructed: it does not leverage a reference set, nor does it incorporate existing uncertainty scores. As a result, unlike our proposed method, it cannot function as a general-purpose tool for transforming heuristic scores into a confidence estimate.

Conformal methods. Conformal prediction (Papadopoulos et al., 2002; Vovk et al., 2005; Angelopoulos & Bates, 2021; Angelopoulos et al., 2024) is a UQ method commonly used in supervised learning whose goal is to output a set of labels which is guaranteed to contain the ground truth label with some user-specified probability, with the size of the produced set quantifying uncertainty. Recent work (Quach et al., 2024; Kladny et al., 2025; Loaiza-Ganem et al., 2026) has leveraged these ideas in the context of LLMs to produce sets of responses. These UQ methods require either a heuristic UQ score or a confidence estimate to obtain their sets; they are thus both orthogonal and compatible with our work.

Motivation. Following the principle that uncertainty scores should be grounded in their usefulness for decision-making (Smith et al., 2025), we observe that while current heuristic scores capture distinct types of uncertainty, they all share the fundamental goal of correlating with response correctness to guide actions like abstention. This observation motivates a framework that can integrate multiple heuristic scores into a single uncertainty metric. We contend that directly predicting the probability of correctness is the most effective way to achieve this; unlike heuristic scores, calibrated probabilities of the outcomes of interest—in this case correctness or lack thereof—provide a principled basis for optimal decision-making. Our first goal, therefore, is to develop a method that can produce a single confidence estimate from one or more pre-existing uncertainty scores, which also offers the additional benefit of being more interpretable than heuristic scores.

We establish two desiderata for our confidence estimates. First, our framework should not introduce requirements beyond those already imposed by its constituent uncertainty scores. For instance, if the pre-existing scores only require black-box access or a single model response, our method should maintain these constraints rather than necessitating white-box access or multiple generations. We will focus on this black-box setting since current state-of-the-art LLMs are all black-box, although our method could also be applied to white-box models. While many current uncertainty scores are zero-shot and require no labelled data, we have already argued that such data is available in any realistic scenario. Our second goal, therefore, is to leverage this available data to produce the best possible confidence estimates from existing scores.

3 Background: Calibration

Consider a target binary variable Y and some features X . A classifier f is said to be calibrated if, for every $p \in [0, 1]$ such that the event $f(X) = p$ is feasible,

$$\mathbb{P}(Y = 1 | f(X) = p) = p. \quad (1)$$

In words, calibration means that the output of the classifier can be trusted as a reliable probability estimate. Note that calibration is distinct from accuracy; for example a classifier might be accurate without being calibrated. Similarly, a calibrated classifier need not be accurate, e.g., the constant classifier $f(x) = \mathbb{P}(Y = 1)$ is perfectly calibrated. Therefore, ideal classifiers should be both accurate and calibrated.

In practice, the calibration of a binary classifier is evaluated through its expected calibration error (ECE). Given a dataset $(x_i, y_i)_{i=1}^n$, the $[0, 1]$ interval is partitioned into L sub-intervals of equal length, I_ℓ , which are then used to obtain data bins, $B_\ell := \{(x_i, y_i) : f(x_i) \in I_\ell\}$. Then,

$$\text{ECE}(f) := \sum_{\ell} \frac{|B_\ell|}{n} |\text{acc}(f, B_\ell) - \text{conf}(f, B_\ell)|, \quad (2)$$

where the sum is over all non-empty bins, $\text{acc}(f, B_\ell)$ is the accuracy of f on bin B_ℓ , and $\text{conf}(f, B_\ell)$ is its confidence on this bin, i.e., its average value.

Many binary classifiers produce a logit $h(x)$, so that f is given by $f(x) = \sigma(h(x))$, where σ denotes the sigmoid function. Temperature scaling is a popular strategy in deep learning (Guo et al., 2017), which modifies a pre-trained classifier by changing its output to $\sigma(h(x)/\tau)$, where $\tau > 0$ is fit to minimize ECE; importantly, the resulting classifier maintains the accuracy of f , but improves its calibration.

4 Method

Rather than proposing a new uncertainty score from first principles, we ask a simpler question:

Given the outputs of existing uncertainty quantification methods and access to labelled evaluation data, what is the best estimate of the probability that an LLM response is correct?

Framing UQ in this way naturally leads to a supervised learning formulation, in which existing uncertainty scores and related statistics are treated as informative features, and the quantity of interest is the probability of correctness itself. We consider a fixed black-box LLM deployed on a task of interest and denote the space of text as $\mathcal{T}$. For a query (prompt) $Q \in \mathcal{T}$, the LLM produces a response $R \in \mathcal{T}$. We denote by $Y \in \{0, 1\}$ the corresponding binary correctness label, i.e., $Y = 1$ if and only if R is a correct response to Q . For a fixed query-response pair, (q, r) , our goal is to estimate the probability of correctness,

$$\eta(q, r) := \mathbb{P}(Y = 1 | Q = q, R = r), \quad (3)$$

which we refer to as the *confidence* of the LLM response.

Available data. As mentioned in Section 2, many heuristic UQ scores generate multiple responses $\mathbf{R} := (R^{(1)}, \dots, R^{(m)}) \in \mathcal{T}^m$ for a single query Q . Here $Y \in \{0, 1\}$ will correspond to the correctness of the first response, $R^{(1)}$, since we think of all the other responses as auxiliary for the computation of uncertainty scores. The way in which each response is generated can change depending on the underlying uncertainty score. For example, the first response could be generated with a low temperature, and subsequent responses could be generated *i.i.d.* with a higher temperature (as in semantic entropy (Kuhn et al., 2023)), or each one could instead correspond to a paraphrased version of Q . We also highlight that $m = 1$ corresponds to the case where the underlying uncertainty scores require a single generation (e.g., P_{true}). We assume access to a labelled dataset,

$$\mathcal{D} := \{(q_i, \mathbf{r}_i, y_i)\}_{i=1}^n, \quad (4)$$

containing queries $q_i \in \mathcal{T}$, responses $\mathbf{r}_i \in \mathcal{T}^m$, and their correctness label $y_i \in \{0, 1\}$. As discussed earlier, such a dataset is typically available in practice, as it is required to assess the performance of UQ methods

prior to deployment. We partition $\mathcal{D}$ into two disjoint subsets. The first is a training set, denoted $\mathcal{D}_{\text{train}}$, which is used to fit the confidence model. The second is a reference set, denoted $\mathcal{D}_{\text{ref}}$, which is used solely to construct auxiliary features for new queries by summarizing the behavior of the LLM on similar, previously evaluated queries. The reference set plays a distinct role from the training set and is central to our approach.

Baseline uncertainty scores as features. A key novelty of our framework is to treat uncertainty scores as *features* rather than final confidence. In doing so, we do not modify or replace existing UQ methods; instead, we reuse their outputs in a supervised learning formulation. We let

$$\mathbf{S}(q, \mathbf{r}) := (S_1(q, \mathbf{r}), \dots, S_c(q, \mathbf{r})) \in \mathbb{R}^c \quad (5)$$

denote a collection of c pre-existing uncertainty scores, $S_j(q, \mathbf{r}) \in \mathbb{R}$ for $j = 1, \dots, c$, associated with a query q and its responses $\mathbf{r}$. These may include heuristic scores or verbalized confidence estimates. An important consequence of this design is that our method introduces no additional requirements beyond those already imposed by the chosen scores. If the scores are computed from a single model generation ($m = 1$), our method does not require additional sampling. This flexibility allows our framework to act as a general-purpose mechanism for aggregating and refining existing uncertainty estimates, rather than as a competing scoring method.

Reference set and neighbourhood-based features. Existing uncertainty scores use only the information derived from a single query and its associated model outputs, without accounting for how the LLM has performed on similar queries in the past. However, when labelled evaluation data is available, this historical information provides a valuable empirical signal about response correctness that is not captured by standalone scores. To exploit this signal, we leverage the reference set $\mathcal{D}_{\text{ref}}$ and construct neighbourhood-based features that summarize the behavior of the LLM on queries similar to a given input.

Embedding space and similarity. We begin by embedding queries into a shared representation space. Let $\phi : \mathcal{T} \rightarrow \mathbb{R}^d$ be an embedding function mapping queries to a vector space. For two queries $q, q' \in \mathcal{T}$, we define their similarity as the cosine similarity between the embeddings,

$$\text{sim}(q, q') := \frac{\langle \phi(q), \phi(q') \rangle}{\|\phi(q)\|_2 \|\phi(q')\|_2}, \quad (6)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and $\|\cdot\|_2$ the corresponding ℓ_2 -norm. Now, for a query q , let

$$\mathcal{N}_k(q) \subset \mathcal{D}_{\text{ref}} \quad (7)$$

denote the set of its k -nearest neighbours in the reference set under the similarity measure defined above.

Neighbourhood statistics. From $\mathcal{N}_k(q)$, we construct summary features that capture both the LLM’s past behavior on similar queries and the local structure of the query space. In particular, we consider:

- The vector of neighbour correctness values,

$$(y_i)_{i: q_i \in \mathcal{N}_k(q)}, \quad (8)$$

as well as the corresponding empirical mean and standard deviation,

$$\mu_y(q) := \frac{1}{k} \sum_{i: q_i \in \mathcal{N}_k(q)} y_i \quad \text{and} \quad \sigma_y(q) := \left(\frac{1}{k-1} \sum_{i: q_i \in \mathcal{N}_k(q)} (y_i - \mu_y(q))^2 \right)^{1/2}. \quad (9)$$

- The vector of similarities to each neighbour,

$$(\text{sim}(q, q_i))_{i: q_i \in \mathcal{N}_k(q)}, \quad (10)$$

as well as the corresponding empirical mean and standard deviation,

$$\mu_{\text{sim}}(q) := \frac{1}{k} \sum_{q_i \in \mathcal{N}_k(q)} \text{sim}(q, q_i) \quad \text{and} \quad \sigma_{\text{sim}}(q) := \left(\frac{1}{k-1} \sum_{q_i \in \mathcal{N}_k(q)} (\text{sim}(q, q_i) - \mu_{\text{sim}}(q))^2 \right)^{1/2}. \quad (11)$$

While the vectors of individual neighbour labels and similarities contain strictly more information, their empirical means (and/or standard-deviation) provide low-dimensional, low-variance summaries of local correctness and similarity structure. Including both aggregated and instance-level features allows downstream classifiers to trade off bias and variance, and to exploit neighbourhood information effectively across different model classes. We will denote the vector containing all these features—i.e., both instance level correctness labels and neighbour similarities, as well as their empirical means and standard deviations—as $\mathbf{f}_{\text{ref}}(q) \in \mathbb{R}^{2k+4}$.

The use of a reference set in this way allows us to reuse past labelled queries to inform uncertainty estimates for new queries. Intuitively, these neighbourhood-based features provide a local estimate of the LLM’s reliability: if past queries similar to q tended to be answered correctly, it is more likely that the current response is also correct. Likewise, the distances to neighbours reflect how representative the reference points are for the current query. By summarizing both correctness and proximity information, the model gains access to empirical patterns that are not captured by standalone uncertainty scores, making it easier to predict response correctness without altering the LLM itself.

Feature representation. For a given query-responses pair $(q, \mathbf{r})$ we construct the feature vector

$$\mathbf{X}(q, \mathbf{r}) := (\mathbf{S}(q, \mathbf{r}), \mathbf{f}_{\text{ref}}(q)) \in \mathbb{R}^{c+2k+4}. \quad (12)$$

This framework is modular: features can be added or removed depending on availability and constraints. In the absence of a reference set, our method reduces to a supervised aggregation of existing uncertainty scores. We also experimented with augmenting the feature representation using high-dimensional query embeddings, in the spirit of approaches such as the one from Ulmer et al. (2024). However, we did not observe consistent improvements over models based on uncertainty scores and neighbourhood statistics alone. We conjecture that this is due to the high dimensionality of embedding representations, which may require substantially more complex models and larger training sets to reliably extract signals predictive of correctness.

Confidence estimation via supervised learning. We cast confidence estimation as a supervised binary classification problem. Let $f_\theta : \mathbb{R}^{c+2k+4} \rightarrow [0, 1]$ be a probabilistic classifier parameterized by θ . We interpret

$$\hat{\eta}(q, r^{(1)}) := f_\theta(\mathbf{X}(q, \mathbf{r})) \quad (13)$$

as an estimate of $\eta(q, r^{(1)}) = \mathbb{P}(Y = 1 \mid Q = q, R^{(1)} = r^{(1)})$. The classifier is trained on $\mathcal{D}_{\text{train}}$ by minimizing a proper scoring rule such as the cross entropy loss,

$$\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(q_i, \mathbf{r}_i, y_i) \in \mathcal{D}_{\text{train}}} [-y_i \log f_\theta(\mathbf{x}_i) - (1 - y_i) \log(1 - f_\theta(\mathbf{x}_i))], \quad (14)$$

where $\mathbf{x}_i = \mathbf{X}(q_i, \mathbf{r}_i)$ is the i -th observed feature vector from the training set.

Our approach improves calibration by explicitly learning the conditional probability of correctness given informative features, rather than relying on heuristic scores alone. Even when $\mathbf{S}(q, \mathbf{r})$ is a confidence estimate (e.g., P_{true}), it is generally not calibrated, e.g., LLM are often over-confident in their responses. By contrast, when a probabilistic classifier is trained using a proper scoring rule, the Bayes-optimal predictor satisfies

$$f^*(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x}). \quad (15)$$

Because our feature representation includes the baseline scores and additional neighbourhood-based statistics, the learned predictor conditions on strictly more information than if it was just given the scores. Under mild regularity conditions, conditioning on additional informative variables cannot worsen calibration and can strictly improve it when the added features are predictive of correctness (Murphy, 1973; Dawid, 1982; Gneiting & Raftery, 2007).

Intuitively, the classifier learns how baseline scores should be rescaled and combined for the task at hand, while neighbourhood features provide empirical estimates of local correctness frequencies. As a result, the learned confidence estimates converge toward calibrated probabilities on the data distribution, whereas fixed heuristic scores cannot adapt to dataset-specific biases.

Inference and computational considerations. At inference time, for a given query-responses pair $(q, \mathbf{r})$, producing $\hat{\eta}(q, r^{(1)})$ proceeds in three steps: (i) computing the baseline uncertainty scores $\mathbf{S}(q, \mathbf{r})$; (ii) retrieving the k -nearest neighbours from the reference set $\mathcal{D}_{\text{ref}}$; and (iii) evaluating the trained classifier f_θ on the combined feature vector. For step (i), note that any computational requirements imposed by a score S_j (e.g., multiple generations) are inherited by our method and not introduced by it. Steps (ii) and (iii), as well as training the classifier, incur negligible computational overhead relative to generating LLM responses, meaning that the computational cost of our method is essentially the same as that of the underlying baseline scores.

Auto-ML for classifier selection. As a minimal baseline, we first consider a temperature-scaled version of the verbalized confidence score ($\text{ts-}P_{\text{true}}$), which can be viewed as the simplest post-hoc model for producing calibrated confidence estimates. Specifically, this approach applies a monotone rescaling to the self-reported probability output by the LLM, requiring only a single scalar parameter and no additional features. It represents the least expressive model one can fit while still leveraging labeled data to improve calibration. Beyond this baseline, rather than training a single classifier, we adopt an Auto-ML approach to select the classifier f_θ that performs best for a given task and feature set. All candidate models are trained on the same feature representations and compared using a unified validation protocol.

In our experiments, we consider two classes of probabilistic classifiers. First, we train an ℓ_1 -regularized logistic regression model, which estimates the probability of correctness. The regularization parameter is selected via k -fold cross-validation by maximizing the area under the receiver operating characteristic curve (AUROC). Second, we train a random forest classifier (Breiman, 2001) to capture nonlinear interactions between features, although the framework is fully compatible with other classifiers, including multi-layer perceptrons, gradient-boosted trees, or any off-the-shelf supervised learner. For random forests, we perform a grid search over the number of trees, maximum tree depth, and the minimum number of samples required to split an internal node. Hyperparameters are selected using k -fold cross-validation, again optimizing AUROC. We additionally apply temperature scaling as a post-hoc calibration step, fitting a single scalar temperature on held-out data to rescale the predictions while preserving their ordering.

Once all the models are trained, we select the one having achieved the best AUROC. While the feature construction process remains fixed across tasks, the resulting estimator f_θ can vary substantially depending on the task, dataset, and feature representation. Linear models may suffice when baseline uncertainty scores are already well aligned with correctness, whereas nonlinear models can better exploit complex interactions induced by neighbourhood-based information. The Auto-ML procedure enables task-adaptive confidence estimation without modifying the LLM or the underlying uncertainty scores.

5 Experiments

In this section, we present empirical results demonstrating that our framework consistently improves upon existing UQ scores. We evaluate our method across multiple datasets, training simple classifiers—logistic regression and random forests—on feature sets that include baseline UQ scores, verbalized confidence estimates, and neighbourhood-based correctness/distance statistics. The objective in all experiments is to predict the correctness of the LLM response. Our code is available at https://github.com/layer6ai-labs/improved_llm_confidence.

5.1 Experimental Setup

Datasets. We consider the following datasets: CommonSense QA (CS_QA) (Talmor et al., 2019), Natural Questions (NQ) (Kwiatkowski et al., 2019), SciQ (Welbl et al., 2017), and SimpleQA (Wei et al., 2024). NQ contains real Google search queries with Wikipedia-sourced answers; we randomly sampled 10,000 datapoints from the training split. We also randomly sampled 8,000 queries each from the training splits of SciQ, a 4-option multiple-choice science exam dataset, and CS_QA, a 5-option multiple-choice reasoning benchmark. For SimpleQA, a dataset from OpenAI that measures short-form factuality, we kept the full dataset of 4,326 examples.

Table 1: AUROC (**higher is better**) and ECE (**lower is better**) across methods. For each method family, we report the best score (highest AUROC, lowest ECE), averaged over five independent runs with different random seeds. Semantic UQ scores are included as features in the Auto-ML method.

	AUROC ($\uparrow$)				ECE ($\downarrow$)			
	P_{true}	Semantic	APRICOT	Auto-ML	P_{true}	ts- P_{true}	APRICOT	Auto-ML
GPT4.1								
CS_QA	0.743	0.645	0.494	0.770	0.069	0.023	0.660	0.024
NQ	0.715	0.709	0.457	0.774	0.322	0.016	0.148	0.029
SciQ	0.895	0.653	0.452	0.911	0.021	0.008	0.966	0.008
SimpleQA	0.632	0.769	0.489	0.782	0.325	0.183	0.312	0.040
GPT4.1-mini								
CS_QA	0.697	0.648	0.498	0.741	0.085	0.081	0.584	0.035
NQ	0.703	0.725	0.504	0.762	0.330	0.015	0.011	0.028
SciQ	0.837	0.702	0.456	0.874	0.041	0.005	0.959	0.007
SimpleQA	0.640	0.790	0.412	0.786	0.113	0.111	0.766	0.024
GPT4.1-nano								
CS_QA	0.626	0.693	0.508	0.710	0.106	0.027	0.462	0.024
NQ	0.734	0.751	0.496	0.790	0.304	0.129	0.221	0.032
SciQ	0.769	0.719	0.543	0.843	0.034	0.008	0.915	0.013
SimpleQA	0.644	0.725	0.490	0.755	0.056	0.056	0.943	0.007

LLM models. To evaluate our method, we worked primarily with the GPT4.1 model family, specifically GPT4.1 (flagship), GPT4.1-mini, and GPT4.1-nano. This choice was guided by two factors. (i) Practical applicability: as closed-source black-box models, these represent the most common type of LLM encountered in real-world production environments where internal weights are inaccessible. (ii) Scalability analysis: by testing across three distinct sizes (standard, mini, and nano), we can observe how our method’s performance and reliability scale with model capacity. Although our main objective is to perform UQ for black-box models, we also consider LLaMA4Maverick, an open-source model, as a way to highlight the broad applicability of our procedure.

Evaluation. Below we summarize the heuristic scores that we build upon and compare against, please refer to Appendix A for more details. All methods are evaluated on a shared test set across five random seeds, with performance assessed using average AUROC and ECE.

Confidence UQ scores. We evaluate two variants of the verbalized confidence score (P_{true}): the raw P_{true} used directly as a probabilistic predictor, and a temperature-scaled version (ts- P_{true}) that refines calibration via a single learned scalar. We also evaluate the APRICOT method, and used the default configuration in the code of Ulmer et al. (2024) for training the neural network predicting confidence.

Semantic-based heuristic UQ scores. We compare against three semantic uncertainty methods: semantic entropy (Kuhn et al., 2023), Laplacian (Lin et al., 2024), and kernel entropy (Nikitin et al., 2024). These methods follow a common procedure: for each query, we sample a total of $m = 21$ responses; the first response corresponds to the response whose uncertainty we wish to quantify and it is sampled with greedy decoding, whereas the other 20 responses are sampled at a high temperature ($T = 1$) to induce higher diversity in the generated responses so as to obtain more meaningful UQ scores. For each model and dataset, we report the best performance achieved by these three methods.

Model-based. We evaluate our approach following the Auto-ML framework described in Section 4. Concretely, we train ℓ_1 -regularized logistic regression and random forest models, reporting the best performance achieved between the two. Neighbourhood-based features are constructed using $k = 20$ nearest neighbours from the reference set. Details on the construction of the reference set using determinantal point processes (DPPs) (Kulesza & Taskar, 2011; 2012) are provided in Appendix B.

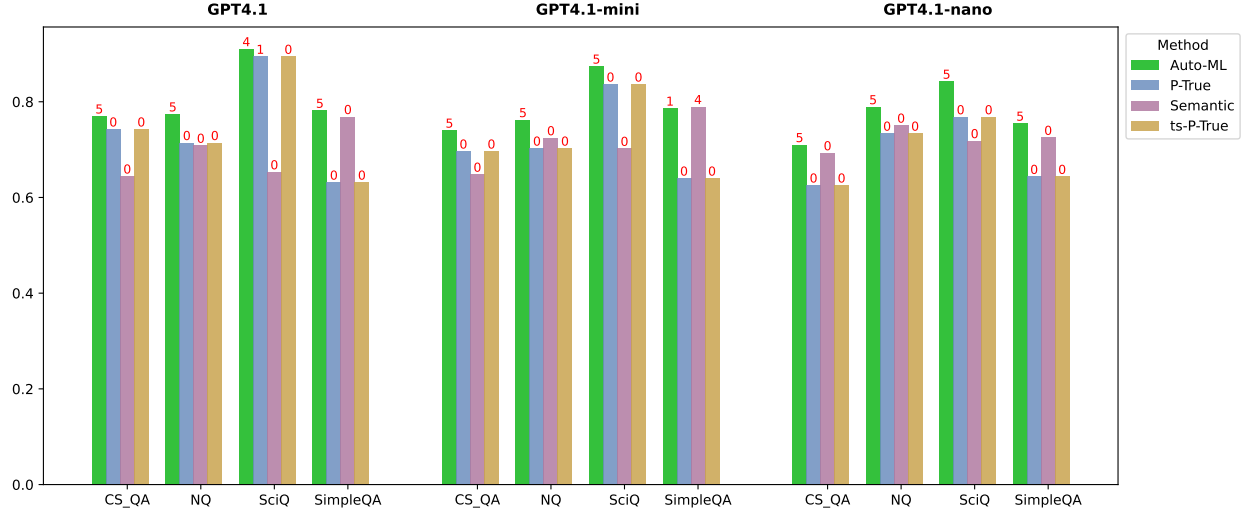


Figure 2: AUROC (**higher is better**) across methods. Each bar shows the average AUROC over five independent runs with different random seeds for each dataset/LLM combination. Red integers above the bars indicate "win" counts, i.e., the number of times a method achieved the highest AUROC among all methods. Semantic UQ scores are included as features in the Auto-ML method.

5.2 Results

Overall, our experimental results demonstrate that leveraging existing uncertainty scores and neighbourhood statistics as features in a simple classifier substantially improves predictive accuracy and yields well-calibrated confidence estimates for LLM responses. This is shown in Table 1, which compares our Auto-ML method when using semantic UQ scores and neighbourhood statistics as features against various baselines. Our proposed Auto-ML procedure consistently outperforms all the baselines in AUROC—there is a single instance where it does not, and the difference is extremely minor. We also note that APRICOT performs quite poorly. Although we could likely improve this baseline by tuning it separately for each task, we highlight that we carried out a fair comparison, as our method involves no tuning. Indeed, avoiding having to tune hyperparameters for the confidence predictor is an important advantage of our method.

Table 1 also reports the best ECE across method families under the same setting; note that we do not compare ECE against semantic methods since they do not provide a confidence score. Our Auto-ML approach consistently beats P_{true} , and while it does not always beat $\text{ts-}P_{\text{true}}$, it is often either a close second or a distant first. We also highlight that $\text{ts-}P_{\text{true}}$ is in a sense the simplest model we can train using an existing score to predict confidence, and so it can also be viewed as an instantiation of our proposed methodology. Overall, Table 1 shows that the confidence scores provided by our method are both highly predictive of correctness and well calibrated. We also emphasize once again that the computational overhead of our Auto-ML confidence over the "Semantic" baseline is negligible.

Figure 2 shows AUROC results and "win" counts across correctness prediction methods for all scenarios. Our Auto-ML method consistently outperforms the baselines. Interestingly, on multiple-choice datasets, semantic entropy-based methods perform poorly, being outperformed by both the Auto-ML method and simple verbalized confidence (P_{true}). More detailed results are provided in Appendix C.

Ablation: changing the model family. Table 2 shows analogous comparisons as those in Table 1 for LLaMA4Maverick: we can see that our Auto-ML method still provides improved UQ scores.

Ablation: removing semantic features. We also evaluate in Table 3 our method using only verbalized confidence (P_{true}) and neighbourhood-based statistics as features, omitting semantic UQ metrics; this re-

Table 2: AUROC (**higher is better**) and ECE (**lower is better**) across methods with LLaMA4Maverick generations.

	AUROC ($\uparrow$)			ECE ($\downarrow$)	
	P_{true}	Semantic	Auto-ML	P_{true}	Auto-ML
LLaMA4Maverick					
NQ	0.727	0.762	0.796	0.077	0.029
SciQ	0.853	0.580	0.857	0.003	0.003
SimpleQA	0.624	0.778	0.787	0.336	0.027

Table 3: AUROC (**higher is better**) and ECE (**lower is better**) across methods **without** semantic features for Auto-ML.

	AUROC ($\uparrow$)			ECE ($\downarrow$)		
	P_{true}	Semantic	Auto-ML	P_{true}	ts- P_{true}	Auto-ML
GPT4.1						
CS_QA	0.743	0.645	0.747	0.069	0.023	0.036
NQ	0.715	0.709	0.735	0.322	0.016	0.025
SciQ	0.895	0.653	0.894	0.021	0.008	0.009
SimpleQA	0.632	0.769	0.667	0.325	0.183	0.088
GPT4.1-mini						
CS_QA	0.697	0.648	0.711	0.085	0.081	0.046
NQ	0.703	0.725	0.721	0.330	0.015	0.062
SciQ	0.837	0.702	0.858	0.041	0.005	0.010
SimpleQA	0.640	0.790	0.649	0.113	0.111	0.076
GPT4.1-nano						
CS_QA	0.626	0.693	0.638	0.106	0.027	0.031
NQ	0.734	0.751	0.753	0.304	0.129	0.114
SciQ	0.769	0.719	0.778	0.034	0.008	0.012
SimpleQA	0.644	0.725	0.662	0.056	0.056	0.033

quires generating only a single response ($m = 1$) and thus has comparable computational cost to P_{true} . Even in this restricted setting, our method achieves strong performance. These results highlight the efficiency of the Auto-ML approach: using only the greedy generation, the verbalized confidence, and a reference set (readily available in practice), the model leverages neighborhood information to produce well calibrated predictions. While this single-sample strategy cannot consistently outperform semantic UQ methods—given the disparity in computational effort and available information—it still delivers competitive AUROC and ECE scores.

5.3 Discussion

Use of the semantic features. The experimental results in Table 1 as well as in Figure 2 show that by leveraging existing UQ methods, we can obtain an accurate and well calibrated predictor of correctness, without additional LLM calls, or access to the LLM’s internal representations. Moreover, in our ablation experiments, we show that: (i) even without the use of semantic features, we can still get a competitive predictor with much lower cost than the semantic methods; and that (ii) our method performs well beyond the black-box GPT family.

Model choice. We use logistic regression and random forests as our classification models due to their low computational overhead and inherent interpretability. These models serve as efficient tools for isolating signals within the feature space and establishing robust performance baselines. While our current implementation focuses on these simpler architectures, the framework is modular. Hence, higher-capacity predictors, such as deep neural networks, could be integrated to potentially capture more intricate, non-linear dependencies in the data.

Limitations. Our approach has two primary limitations. First, it requires training dataset-specific classifiers, which may limit immediate out-of-the-box generalization. However, this is largely mitigated by the fact that most practical applications involve optimizing a model for a specific target domain or task. Second, the method’s predictive accuracy depends on the reference set, as it requires neighbours for each question to extract meaningful local information. Consequently, a sufficiently large dataset is needed to provide high-quality retrieval. In practice, this requirement is typically satisfied during the standard model development lifecycle, where a representative dataset is already a prerequisite for evaluation and fine-tuning.

6 Conclusions and Future Work

In this work, we have demonstrated that confidence prediction for large language models should be treated as a machine learning problem. Our central thesis is that regardless of the initial UQ score, performance can almost always be improved by utilizing those scores as features within a supervised framework. While we used straightforward models like logistic regression and random forests to establish a baseline, the integration of more sophisticated Auto-ML pipelines is bound to further refine these predictions and yield even higher calibration and discriminative power.

Our advocacy for this classification-based approach is not a dismissal of research into heuristic UQ scores. On the contrary, we view heuristic scores as essential inputs, and our framework provides a systematic way of enhancing them by learning how to weigh and combine different signals effectively. Finally, while this paper focused on binary correctness, future research should expand this paradigm toward multi-class classification. Moving beyond a simple “correct / incorrect” binary allows for a more nuanced evaluation of model outputs by distinguishing between responses that are outright incorrect, partially correct, acceptable, or excellent, thereby providing a more robust foundation for reliable deployment LLMs.

References

- Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv:2107.07511*, 2021.
- Anastasios N Angelopoulos, Rina Foygel Barber, and Stephen Bates. Theoretical foundations of conformal prediction. *arXiv:2411.11824*, 2024.
- Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. In *Nature Machine Intelligence*, volume 1, pp. 20–23, 2019.
- James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.
- Leo Breiman. Random forests. In *Machine learning*, volume 45, pp. 5–32. Springer, 2001.
- Rijul Chaturvedi and Sanjeev Verma. *Opportunities and Challenges of AI-Driven Customer Service*, pp. 33–71. Springer International Publishing, 2023.
- Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31, 2018.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder,

-
- Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgren Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.
- A Philip Dawid. The well-calibrated bayesian. *Journal of the American statistical Association*, 77(379): 605–610, 1982.
- Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5050–5063, 2024.
- Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. SPUQ: Perturbation-based uncertainty quantification for large language models. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2024.
- Tilman Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.
- Carlos Gómez-Rodríguez and Paul Williams. A Confederacy of Models: a Comprehensive Evaluation of LLMs on Creative Writing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14504–14528, 2023.
- Yashvir S Grewal, Edwin V Bonilla, and Thang D Bui. Improving uncertainty quantification in large language models via semantic embeddings. *arXiv:2410.22685*, 2024.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, 2017.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.
- Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. Decomposing uncertainty for large language models through input clarification ensembling. In *International Conference on Machine Learning*, 2024.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv:2207.05221*, 2022.
- Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. Calibration-tuning: Teaching large language models to know what they don’t know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP*, pp. 1–14, 2024.
- Michael Kirchhof, Gjergji Kasneci, and Enkelejd Kasneci. Position: Uncertainty quantification needs reassessment for large language model agents. In *International Conference on Machine Learning*, 2025.

-
- Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. Conformal generative modeling with improved sample efficiency through sequential greedy filtering. In *International Conference on Learning Representations*, 2025.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023.
- Alex Kulesza and Ben Taskar. K-DPPs: Fixed-Size Determinantal Point Processes. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1193–1200, 2011.
- Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL <https://aclanthology.org/Q19-1026/>.
- Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. In *Transactions on Machine Learning Research*, 2024.
- Gabriel Loaiza-Ganem, Kevin Zhang, Wei Cui, Marc T Law, and Kin Kwan Leung. Conf-gen: Conformal uncertainty quantification for generative models. In *International Conference on Machine Learning*, 2026.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1906–1919, 2020.
- Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents’ overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.
- Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.
- Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *Advances in Neural Information Processing Systems*, 2024.
- Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, 2002.
- Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *Advances in Neural Information Processing Systems*, 2024.
- Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024.
- Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018.
- Brendan Leigh Ross, Noël Vouitsis, Atiyeh Ashari Ghomi, Rasa Hosseinzadeh, Ji Xin, Zhaoyan Liu, Yi Sui, Shiyi Hou, Kin Kwan Leung, Gabriel Loaiza-Ganem, and Jesse C Cresswell. Textual Bayes: Quantifying uncertainty in LLM-based systems. *arXiv:2506.10060*, 2025.
- Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

-
- Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*, 2025.
- Freddie Bickford Smith, Jannik Kossen, Eleanor Trollope, Mark van der Wilk, Adam Foster, and Tom Rainforth. Rethinking aleatoric and epistemic uncertainty. In *International Conference on Machine Learning*, 2025.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 2019. URL <https://arxiv.org/abs/1811.00937>.
- Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Oh. Calibrating large language models using their generations only. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 15440–15459, 2024.
- Noël Vouitsis, Zhaoyan Liu, Satya Krishna Gorti, Valentin Villecroze, Jesse C Cresswell, Guangwei Yu, Gabriel Loaiza-Ganem, and Maksims Volkovs. Data-efficient multimodal fusion on a single gpu. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27239–27251, 2024.
- Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, 2005.
- Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, {Carla P.} Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, {Tie Yan} Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, {Connor W.} Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. In *Nature*, volume 620, pp. 47–60, 2023a.
- Xi Wang, Laurence Aitchison, and Maja Rudolph. LoRa ensembles for large language model fine-tuning. *arXiv:2310.00035*, 2023b.
- Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Conference on Empirical Methods in Natural Language Processing*, 2021.
- Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. 2024. URL <https://arxiv.org/abs/2411.04368>.
- Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. 2017. URL <https://arxiv.org/abs/1707.06209>.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv:2401.11817*, 2024.
- Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *International Conference on Learning Representations*, 2024a.
- Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv:2412.14737*, 2024b.
- Kaitlyn Zhou, Jena D Hwang, Xiang Ren, and Maarten Sap. Relying on the unreliable: The impact of language models’ reluctance to express uncertainty. *arXiv:2401.06730*, 2024.

A Heuristic Scores Based on Multiple Generations

Here we summarize the heuristic scores based on multiple generated responses that we use in Section 5. Throughout this section, we consider a query q and m generated responses to it, $r^{(1)}, \dots, r^{(m)}$. We use $m = 21$ in all our experiments, and use the last $m - 1 = 20$ responses, $r^{(2)}, \dots, r^{(21)}$, to compute the scores described below. Here we only describe how these scores are calculated, we refer the reader to the corresponding references for the intuitions behind these calculations.

Semantic Entropy (Kuhn et al., 2023). To compute semantic entropy, the $m - 1$ responses are first clustered into semantically equivalent groups; we follow Kuhn et al. (2023) and do this by using an entailment model (He et al., 2021), which given two responses, classifies the first one as entailing, or not entailing, the second one. The entailment model is called on every pair of responses, producing a matrix $K \in \mathbb{R}^{(m-1) \times (m-1)}$, where K_{ij} contains the probability that the entailment model assigns to $r^{(i+1)}$ entailing $r^{(j+1)}$ for $i, j = 1, \dots, m - 1$. Applying a greedy algorithm on K then produces the clusters. Once these clusters are obtained, we compute the relative frequency $\hat{p}_g$ for every cluster g , and then semantic entropy is computed as

$$-\sum_g \hat{p}_g \log \hat{p}_g. \quad (16)$$

Laplacian Uncertainties (Lin et al., 2024). We also compute two scores proposed by Lin et al. (2024). These scores are based on the same matrix K which we computed above for semantic entropy. First, K is symmetrized to obtain

$$W := \frac{K + K^\top}{2}. \quad (17)$$

Then, the matrix D is defined as a diagonal matrix, whose i -th diagonal entry is given by the sum of the i -th row of W . A final matrix is then computed,

$$L := I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}. \quad (18)$$

From this matrix both metrics can be computed. The one we label ‘‘Laplacian Degree’’ is given by

$$1 - \frac{\text{trace}(L)}{(m - 1)^2}, \quad (19)$$

and the one we label ‘‘Laplacian Eigenvalue’’ is given by

$$\sum_i \max(0, 1 - \lambda_i(L)), \quad (20)$$

where $\lambda_i(L)$ is the i -th eigenvalue of L .

Kernel Entropy Uncertainty (Nikitin et al., 2024). The last score we consider is also based on the matrix K . First, the matrix

$$L' := D - W \quad (21)$$

is obtained, from which we compute

$$A := e^{-0.4L'}. \quad (22)$$

Then, defining D' as the diagonal matrix whose i -th diagonal entry is $1/\sqrt{A_{ii}}$, we compute a final matrix,

$$B := \frac{1}{m - 1} D' A D'. \quad (23)$$

The final score, which we label ‘‘Von Neumann Entropy’’, is given by

$$-\sum_i \lambda_i(B) \log \lambda_i(B). \quad (24)$$

Note that the main computational bottleneck to compute these scores is the generation of $m - 1$ responses. Once this is done, computing K , and then each of the scores, is quick. This observation further motivates our approach: it makes sense to combine all these scores into a single confidence score, since if we have already computed one, we might as well compute the others too.

B Determinantal Point Processes

A natural way to obtain the reference dataset from $\mathcal{D}$ is to simply sample elements uniformly at random, without replacement, to obtain $\mathcal{D}_{\text{ref}}$. However, intuitively, we would want $\mathcal{D}_{\text{ref}}$ to be as diverse as possible, so that the features we construct from it will always be as informative as possible. DPPs (Kulesza & Taskar, 2012) are a way of randomly sampling diverse subsets of data. DPPs require a similarity matrix S , where S_{ij} measures the similarity between datapoint i and datapoint j . By leveraging this matrix, DPPs produce a random subset of the given dataset, but the resulting subset is not generated through uniform sampling without replacement; rather, it assigns higher probability to diverse (as measured by S) subsets. We instantiated S as

$$S_{ij} = (1 + \text{sim}(q_i, q_j))^2,$$

and we used k -DPPs (Kulesza & Taskar, 2011) rather than the standard DPPs, as this choice allows to specify the size of the resulting subset in advance. We follow Vouitsis et al. (2024) and use the algorithm proposed by Chen et al. (2018), which outputs a greedy estimate of the mode of the k -DPP: this can be understood as a “maximally diverse” subset of a specified size.

Overall, k -DPPs do not produce a large improvement over simply sampling the reference dataset uniformly at random. That being said, we must compute all the embeddings anyway for feature construction (which is the main computational bottleneck for using DPPs in the first place), so that the overhead of using k -DPPs is minimal. Since we did see a small improvement in performance from k -DPPs, we thus decided to use them.

C Additional Results

For completeness, we provide detailed AUROC results for all experiments across the three LLMs and all datasets and methods considered. Table 4 reports the mean and standard deviation of AUROC over five independent random seeds for each combination with GPT4.1 generations. Table 5 displays the AUROC for each method with GPT4.1-mini generations, Table 6 displays the AUROC for each method with GPT4.1-nano generations, and Table 7 for LLaMA4Maverick.

Table 4: Mean (standard deviation) AUROC of the different methods and datasets with **GPT4.1** generations.

Method	CS_QA	NQ	SciQ	SimpleQA
P_{true}	0.743 (0.024)	0.715 (0.008)	0.895 (0.011)	0.632 (0.009)
Logistic Regression	0.764 (0.023)	0.747 (0.008)	0.903 (0.011)	0.780 (0.023)
Random Forest	0.770 (0.020)	0.774 (0.008)	0.906 (0.023)	0.779 (0.022)
Laplacian Degree	0.644 (0.011)	0.697 (0.006)	0.564 (0.035)	0.766 (0.016)
Laplacian Eigenvalue	0.630 (0.007)	0.708 (0.013)	0.653 (0.011)	0.766 (0.016)
Semantic Entropy	0.614 (0.014)	0.646 (0.009)	0.625 (0.023)	0.754 (0.019)
Von Neumann Entropy	0.644 (0.011)	0.698 (0.006)	0.564 (0.035)	0.766 (0.017)

Table 7: Mean (standard deviation) AUROC of the different methods and datasets with **LLaMA4Maverick** generations.

Method	NQ	SciQ	SimpleQA
P_{true}	0.727(0.010)	0.853(0.037)	0.624(0.019)
Logistic Regression	0.792(0.010)	0.817(0.078)	0.787(0.010)
Random Forest	0.796(0.010)	0.842(0.038)	0.771(0.017)
Laplacian Degree	0.758(0.006)	0.476(0.037)	0.776(0.013)
Laplacian Eigenvalue	0.762(0.008)	0.550(0.075)	0.768(0.006)
Semantic Entropy	0.725(0.005)	0.535(0.023)	0.746(0.012)
Von Neumann Entropy	0.758(0.006)	0.476(0.037)	0.778(0.014)

Table 5: Mean (standard deviation) AUROC of the different methods and datasets with **GPT4.1-mini** generations.

Method	CS_QA	NQ	SciQ	SimpleQA
P_{true}	0.697 (0.019)	0.703 (0.018)	0.837 (0.033)	0.640 (0.026)
Logistic Regression	0.736 (0.025)	0.754 (0.008)	0.869 (0.028)	0.786 (0.027)
Random Forest	0.737 (0.030)	0.762 (0.013)	0.874 (0.027)	0.774 (0.035)
Laplacian Degree	0.646 (0.025)	0.721 (0.005)	0.622 (0.057)	0.788 (0.022)
Laplacian Eigenvalue	0.631 (0.029)	0.722 (0.006)	0.699 (0.046)	0.785 (0.024)
Semantic Entropy	0.622 (0.027)	0.691 (0.005)	0.674 (0.031)	0.778 (0.025)
Von Neumann Entropy	0.646 (0.025)	0.723 (0.005)	0.622 (0.057)	0.788 (0.021)

Table 6: Mean (standard deviation) AUROC of the different methods and datasets with **GPT4.1-nano** generations.

Method	CS_QA	NQ	SciQ	SimpleQA
P_{true}	0.626 (0.010)	0.734 (0.015)	0.769 (0.017)	0.644 (0.020)
Logistic Regression	0.707 (0.009)	0.778 (0.009)	0.838 (0.024)	0.748 (0.035)
Random Forest	0.709 (0.010)	0.790 (0.011)	0.838 (0.029)	0.745 (0.044)
Laplacian Degree	0.684 (0.019)	0.751 (0.013)	0.668 (0.041)	0.703 (0.033)
Laplacian Eigenvalue	0.691 (0.015)	0.741 (0.013)	0.716 (0.038)	0.718 (0.038)
Semantic Entropy	0.660 (0.008)	0.706 (0.014)	0.708 (0.026)	0.725 (0.040)
Von Neumann Entropy	0.684 (0.019)	0.750 (0.013)	0.668 (0.041)	0.701 (0.031)